

FIG. 1

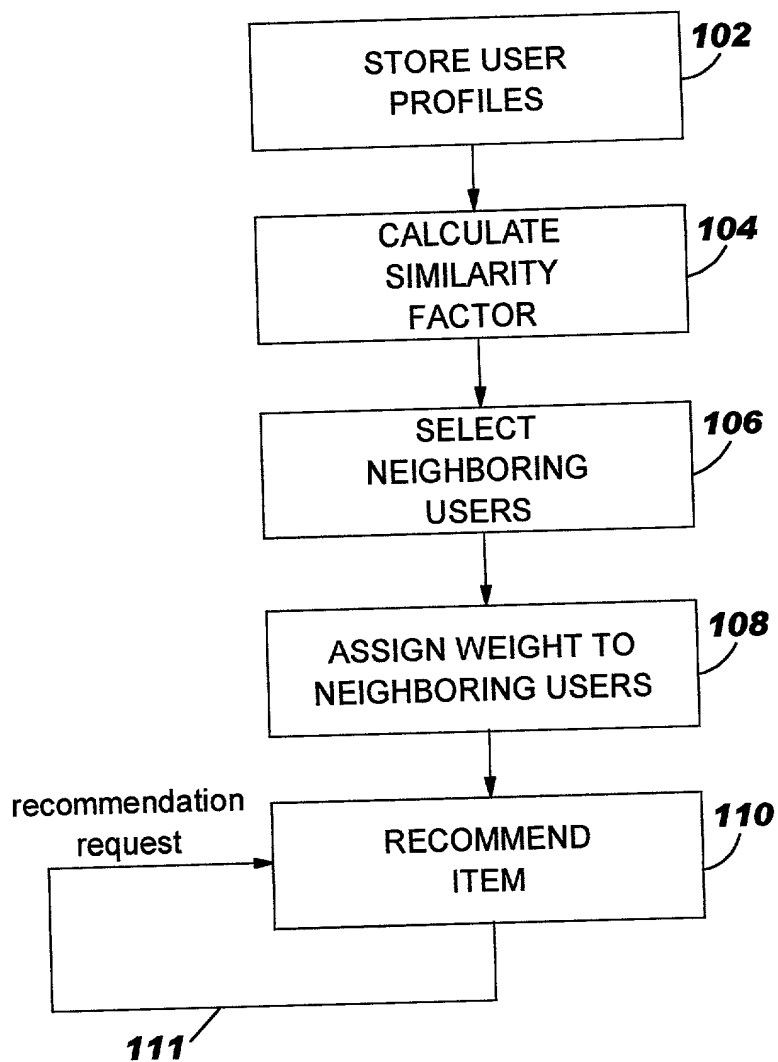
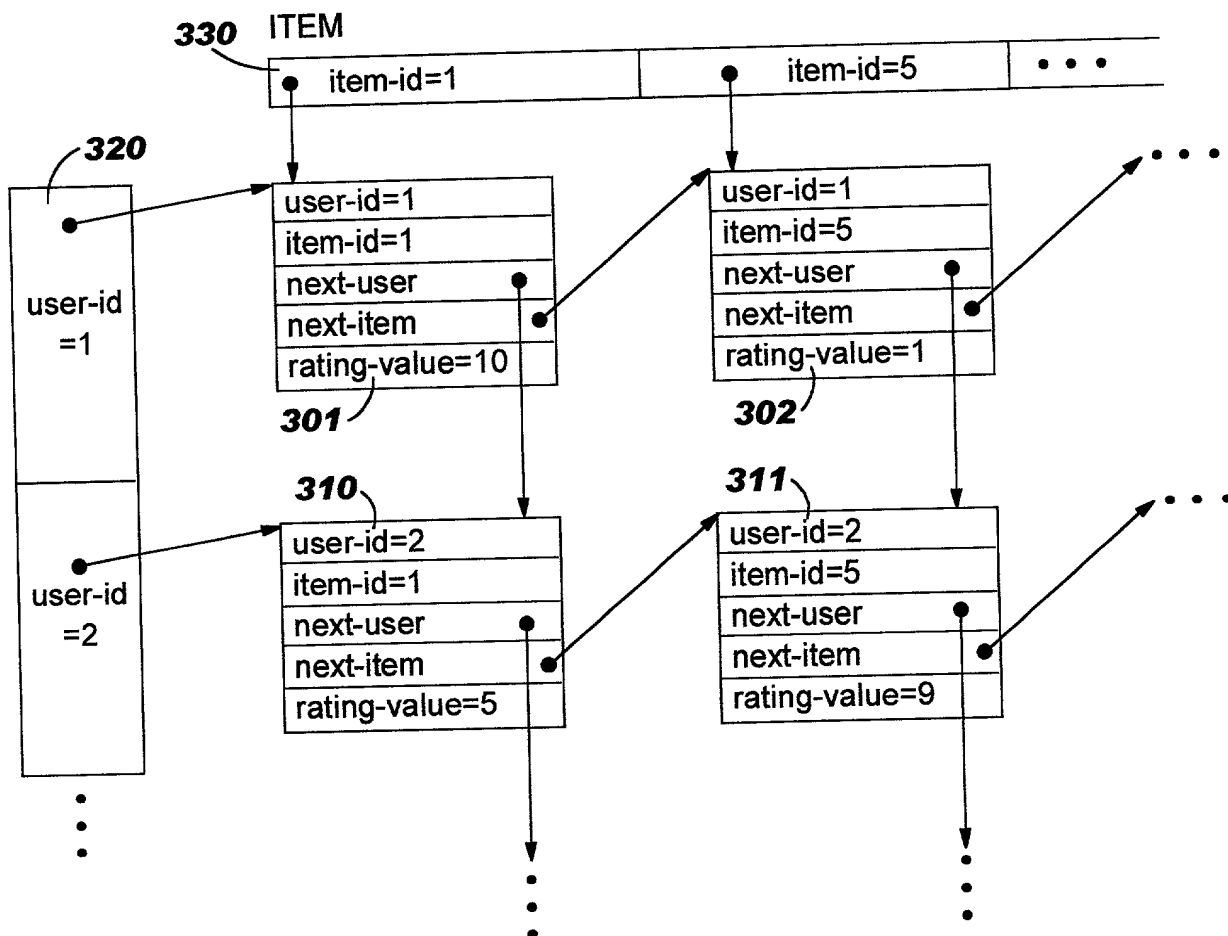


FIG. 2

|              |
|--------------|
| user-id      |
| item-id      |
| next-user    |
| next-item    |
| rating-value |

FIG. 3



## FIG. 4

```
401.calc(u,selected)
402.define used[v]=false for each user v
403.used[u]=true
404.initialize list N as empty list
405.
406.foreach rating ru of list USER[u] do
407.  if (selected[ru.item-id])
408.    foreach rating ri of list ITEM[ru.item-id] do
409.      u'=ri.user-id
410.      if (used[u']==false
411.        used[u']=true
412.        append tuple (u',similarity)u,u',selected)) to list N
413.
414.sort N by value t.s of each tuple t=(t.u,t.s) by quicksort/heapsort
415.
416.return N

420.similarity(x,y,selected)
421.this function is returning the similarity between user x and user y,
    eg. the Pierce coefficient.
422.It will be computed only on the items "it" for which selected[it]==true
```

## FIG. 5

```
501.calc(u,selected)
502.if there is no computed list N(u) for user u just do the "normal" calc(u,selected)
503.
504.if timestamp(last update u) > timestamp(N(u)) do the "normal" calc(u,selected)
505.
506.foreach tuple t=(t.u,t.s) of N(u) do
507.    if (timestamp(last update t.u) > timestamp(N(u))
508.        t.s = similarity(u,t.u,selected)
509.
510.sort N(u) by bubble sort if the number of updates is small,
    otherwise by quicksort/heapsort
511.update timestamp of N(u)
512.return N(u)
```